NASA Technical Memorandum 104206
AVSCOM Technical Report 92-B-007

OBSERVATIONS ON COMPUTATIONAL METHODOLOGIES FOR USE IN LARGE-SCALE, GRADIENT-BASED, MULTIDISCIPLINARY DESIGN INCORPORATING ADVANCED CFD CODES

P. A. Newman
G. J.-W. Hou
H. E. Jones
A. C. Taylor III
V. M. Korivi

February 1992

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, VA 23665-5225

US ARMY
AVIATION
SYSTEMS COMMAND
AVIATION R&T ACTIVITY

OBSERVATIONS ON COMPUTATIONAL METHODOLOGIES FOR USE IN LARGE-SCALE, GRADIENT-BASED, MULTIDISCIPLINARY DESIGN INCORPORATING ADVANCED CFD CODES

P. A. Newman
NASA Langley Research Center, Hampton, VA

G. J-. W. Hou
Old Dominion University, Norfolk, VA

H. E. Jones
USA-AVSCOM, Hampton, VA

A. C. Taylor III and V. M. Korivi
Old Dominion University, Norfolk, VA

ABSTRACT

This paper briefly outlines how a combination of various computational methodologies could reduce the enormous computational costs envisioned in using advanced CFD codes in gradient-based optimized multidisciplinary design (MdD) procedures. Implications of these MdD requirements upon advanced CFD codes are somewhat different than those imposed by a "single" discipline design. A means for satisfying these MdD requirements for gradient information is presented which appears to permit: first, some leeway in the CFD solution algorithms which can be used; second, an extension to 3-D problems; and third, straightforward use of other computational methodologies. Many of these observations have previously been discussed as possibilities for doing parts of the problem more efficiently; the contribution here is observing how they fit together in a mutually beneficial way.

INTRODUCTION

A number of observations concerning various computational methodologies relevant to large-scale, multidisciplinary, gradient-based optimization for engineering systems design problems are discussed. In particular, these observations address the situation where one (or more) discipline response(s) required by the optimized design procedure involves the analysis of a system of nonlinear partial differential equations (PDE) which, when discretized, form a very large set of algebraic equations which must be solved iteratively. This situation occurs, for example, when advanced computational fluid dynamics (CFD) codes are applied in a multidisciplinary sensitivity analysis (SA) procedure for optimizing an aerospace vehicle design. This is the context in which the following remarks are cast; however, it is felt that the present discussion and observations have broader utility and more general applications.

This paper discusses these observations in a general way, briefly outlining how the combination of various computational methodologies could reduce the enormous computational costs envisioned in using advanced CFD codes in an optimized multidisciplinary design (MdD) procedure. Implications of these MdD requirements upon advanced CFD codes are somewhat different than those imposed by a "single" discipline design. A means for satisfying these MdD requirements for gradient information is presented which appears to permit: first, some leeway in the CFD solution algorithms which can be used; second, an extension to 3-D problems; and third, straightforward use of other computational methodologies. Many of these observations have previously been discussed as possibilities for doing parts of the problem more efficiently; the contribution here is observing how they fit together in a mutually beneficial way.

The present interest and work have been stimulated by two research programs related to incorporating advanced CFD capabilities in MdD. The NASA Langley Research Center High-Speed Airframe Integration Research (HiSAIR) project (Refs.[1, 2]) is focused on the High-Speed Civil Transport (HSCT) design activity in order to develop methodology and computational environment for MdD and analysis. The emphasis is on including most of the required disciplines and interactions at a sufficiently advanced level of analysis in order to demonstrate improved engineering designs. The second stimulus is the NASA Computational Aerosciences (CAS) project of the High Performance Computing and Communications (HPCC) Program (Ref.[3]) where one of the applications is the HSCT. The two major thrusts are enhanced simulations via multidisciplinary formulations and improved computational efficiency via massively parallel hardware. In both programs, the primary NASA Langley approach being pursued is MdD via sensitivity analysis.

MULTIDISCIPLINARY DESIGN REQUIREMENTS

A recent survey by Jameson (Ref.[4]) discusses the successes and challenges in CFD and outlines numerical algorithms currently being used to obtain iterative solutions for the various systems of type-dependent, nonlinear PDE's mathematically modelling different levels of physical approximation and configuration complexity. Automated optimization in aerodynamic design is listed as one of the remaining challenges for CFD. He primarily discusses aerodynamic design and shape optimization as "single" discipline design problems. These problems continue to be topics of much interest as evidenced, for example, in the survey by Dulikravich (Ref.[5]) where a number of methods are discussed; and also in the recent works by Jameson (Refs.[6, 7]) and Ta'asan, et al. (Ref.[8]) which utilize adjoint approaches. In these latter papers, the adjoint variable is formed with respect to a cost function generally composed of both objective and constraint functions appropriate to a single discipline. For a single discipline design, these latter design methods appear to be computationally much more efficient and therefore more feasible than the optimizer-based SA methods.

Procedures for multidisciplinary optimization of engineering systems have been addressed by Sobieski (Ref.[9]). He proposes a unified system SA guided by system sensitivity derivatives (SD); the optimizer code or algorithm which uses these SD is the outer-most loop of the entire design process. The objective and constraint functions are now generally composed of output functions from several disciplines. Each single-discipline analysis code is then to supply not only the output functions required for the constrained optimization process and other discipline analysis inputs, but also the partial derivatives of all of these output functions with respect to its input variables. These variables include not only the MdD variables, but also output functions from other disciplines which implicitly depend on the MdD variables. However, the application of advanced CFD to provide aerodynamic analyses within such procedures appears to be severely hampered by the sheer magnitude of the computational tasks involved, particularly if these SD must be obtained by "brute-force" finite differencing. There has been recent interest and progress in obtaining quasi-analytical SD from advanced CFD codes as evidenced in the works by Elbanna & Carlson (Ref.[10]); Taylor, et al. (Ref.[11]); Baysal, et al. (Ref.[12]); and Shubin & Frank (Ref.[13]), for example. As can be seen, direct solver methods have been used in 2–D problems; however, their use in 3–D optimized MdD via SA appears highly unlikely (Ref.[10]) as a viable approach. In the appendix of this paper, an estimate (perhaps optimistic) of the magnitude of the advanced CFD task for steady external aerodynamics in 3–D MdD via SA is given.

IMPLICATIONS OF MdD REQUIREMENTS FOR CFD[1]

In the system SA procedure of Sobieski (Ref.[9]), a CFD code must be able to compute efficiently either the sensitivity functions, Q' (taken herein to mean $\frac{\partial Q}{\partial D}$), of its solution state vector, Q, with respect to each of its own input design variables, D, or else, the adjoint variables, A (constructed with respect to each of its required output functions, F) in order that the required output function SD can be formed. One concludes from the work of Shubin & Frank (Ref.[13]) that the sensitivity functions Q' or adjoint variables A must be consistent discrete derivatives or adjoints of the discrete solution state vector Q in order for the optimization procedure to be fully

[1] In this and following sections, dependencies of quantities upon grid point indices, discipline index, unknown variable index, design variable index, output function index, etc. have been suppressed. All equations are vector or matrix equations, generally with very large dimensions. These indicial dependencies and dimensional estimates are given in the Appendix.

successful, i.e., converge. This aspect appears to adversely affect direct application to advanced CFD codes; the discretized systems of equations for Q' or A are not as well conditioned as in other disciplines and additional errors, as noted below, may occur. Aspects of retrofitting existing CFD codes to compute SD have been discussed in a recent report by Shubin (Ref.[14]).

In advanced CFD codes, the steady flow-field solution state vector Q is determined from a set of discretized nonlinear algebraic equations which are solved iteratively. Symbolically, $R(Q, D, X) = 0$ where the dependence of the residual R on the discrete grid X is also noted. Iteration according to Newton's method (superscript n denotes iteration step):

$$\begin{gathered} -\frac{\partial R^n}{\partial Q}\Delta Q^n = R^n(Q^n, D, X) \\ Q^{n+1} = Q^n + \Delta Q^n \end{gathered} \tag{1}$$

gives $Q^\star$, the "root" of $R(Q^\star, D, X) = 0$ at convergence. The output functions, F, required by MdD (and analysis) consist, for example, of quantities like surface loadings (for structures), lift-drag-moment polars (for performance), stability derivatives (for control), etc. at multiple flow conditions. Most of these functions then are taken as $F = F(Q^\star, D, X)$ where $Q^\star$ and X are usually evaluated at the configuration surface; that is, on the boundary.

The SD of a required output F with respect to an input D is then

$$\frac{dF}{dD} = \frac{\partial F}{\partial D} + \frac{\partial F}{\partial Q}Q' + \frac{\partial F}{\partial X}X' \tag{2}$$

Thus one needs partial derivatives (sensitivity functions) such as Q' and X' (taken herein to mean $\frac{\partial X}{\partial D}$). Note that for the "body-oriented" or surface fitted grids of modern CFD codes, $X' \neq 0$ (it may not even be sparse) since the surface shape design variables are included in the set D and are "boundary-values" for the grid generation equations or problem. Some aspects related to obtaining these derivatives are discussed in Refs.[15] and [16]. All derivatives in Eq. (2) can be generated by "brute-force" finite differencing; but, there are drawbacks, not the least of which is an inordinate computational task. If one applies the quasi-analytical approach suggested by Sobieski (Ref.[17]), then from $R^\star(Q, D, X) = 0$ one obtains upon differentiating with respect to D (recognizing that $\frac{dR^\star}{dD} = 0$):

$$-\frac{\partial R^\star}{\partial Q}Q' = \frac{\partial R^\star}{\partial D} + \frac{\partial R^\star}{\partial X}X' \tag{3}$$

where superscript $\star$ means evaluated at $Q^\star$. This is essentially the linear equation from which most CFD solutions for Q' have been obtained via either direct solver or iterative methods. See, for example, (Refs.[10–13]) and the earlier references quoted by them. Comparison of Eqs. (1) and (3) shows that if a Newton solver is used to get $Q^\star$, then sensitivity functions, Q' , come essentially free for new right hand sides since $\frac{\partial R^\star}{\partial Q}$ is already known and factored. This has been previously noted and used in 2-D problems. For 3-D CFD problems, it appears that iterative solutions must be used to get Q' because the direct solvers require too much memory; however, as noted in Ref.[10], many iterative algorithms for solving Eq. (3) converge very slowly. This and the concerns expressed by Shubin (Ref.[14]) prompted the present approach for obtaining consistent SD.

AN INCREMENTAL ITERATIVE FORM FOR CONSISTENT SD

A closer comparison of the nonlinear Eq. (1) used to obtain the solution Q and typical iterative forms of the linear Eq. (3) used to obtain the sensitivity Q', however, shows another fundamental difference between the two. Eq. (1) is written in the incremental or "delta-" form (unknown is ΔQ^n) and the RHS vanishes when the residual R vanishes; i.e., when the implicit discretized state equations for steady flow are satisfied. The operator on the LHS needs not be exactly $\frac{\partial R^n}{\partial Q}$ in order to obtain convergence; in fact, for almost all 3-D CFD algorithms the operator is not the Newton operator because there are truncation, iterative, and/or splitting errors. Acceleration of an iterative solution for $Q^\star$ is obtained from Eq. (1) by modifying the LHS operator (See, for example, Ref.[4], where Jameson calls this the "correction" form).

The iterative incremental form of Eq. (3) for the sensitivity Q' is

$$
\begin{gathered}
-\widetilde{\frac{\partial R^\star}{\partial Q}}(\Delta Q')^n = \frac{\partial R^\star}{\partial Q}(Q')^n + \frac{\partial R^\star}{\partial D} + \frac{\partial R^\star}{\partial X}X' = \left(\frac{dR^\star}{dD}\right)^n \\
(Q')^{n+1} = (Q')^n + (\Delta Q')^n
\end{gathered}
\tag{4}
$$

where $\widetilde{\frac{\partial R^\star}{\partial Q}}$ implies that the LHS operator need not be exactly $\frac{\partial R^\star}{\partial Q}$. All that is required here is a LHS operator approximation with good convergence properties; when the RHS vanishes ($\frac{dR^\star}{dD}$ is zero), consistent discrete derivatives in the sense of Ref.[13] will be obtained if the discrete approximation on the RHS of Eq. (4) is consistent with that on the RHS of Eq. (1). The detailed computational aspects of and advantages from using Eq. (4) rather than Eq. (3) to obtain SD for CFD solutions are discussed by Korivi, et al. (Ref.[18]).

Similar comments apply to the discrete adjoint variables A, constructed with respect to the required output functions F. Following the procedure of Hou, et al. (Ref.[19]) an adjoint variable A for each function F is introduced as the multiplier of $R^\star(Q,D,X) = 0$ and then added to F so that formally one obtains

$$
\tilde{F}(Q,D,X) = F(Q,D,X) + A^T R^\star(Q,D,X) \tag{5}
$$

where superscript T means transpose and the tilde indicates a functional difference between the two F's. Differentiation with respect to the design variable D can be written several ways

$$
\begin{aligned}
\frac{d\tilde{F}}{dD} &= \frac{dF}{dD} + A^T\frac{dR^\star}{dD} \\
&= \frac{\partial \tilde{F}}{\partial D} + \frac{\partial \tilde{F}}{\partial X}X' + \frac{\partial \tilde{F}}{\partial Q}Q' \\
&= \frac{\partial F}{\partial D} + \frac{\partial F}{\partial X}X' + A^T\left(\frac{\partial R^\star}{\partial D} + \frac{\partial R^\star}{\partial X}X'\right) + \left(A^T\frac{\partial R^\star}{\partial Q} + \frac{\partial F}{\partial Q}\right)Q'
\end{aligned}
\tag{6}
$$

The discrete adjoint variable A is defined as the solution of $\left(\frac{\partial \tilde{F}}{\partial Q}\right)^T = 0$; that is,

$$
-\left(\widetilde{\frac{\partial R^\star}{\partial Q}}\right)^T A = \left(\frac{\partial F}{\partial Q}\right)^T \tag{7}
$$

Thus the coefficient of Q' in Eq. (6) vanishes, so that

$$\frac{d\tilde{F}}{dD} = \frac{\partial F}{\partial D} + \frac{\partial F}{\partial X}X' + A^T\left(\frac{\partial R^\star}{\partial D} + \frac{\partial R^\star}{\partial X}X'\right) \tag{8}$$

The incremental form for iteratively solving Eq. (7) for the discrete adjoint A is

$$-\widetilde{\left(\frac{\partial R^\star}{\partial Q}\right)}^T \Delta A^n = \left(\frac{\partial R^\star}{\partial Q}\right)^T A^n + \left(\frac{\partial F}{\partial Q}\right)^T = \left(\left(\frac{\partial \tilde{F}}{\partial Q}\right)^T\right)^n \tag{9}$$
$$A^{n+1} = A^n + \Delta A^n$$

where here again, the tilde on the LHS indicates that the discrete operator may be approximate. Upon convergence, the RHS is zero; that is, Eq. (7) is satisfied. Therefore, the RHS must be a consistent discrete form agreeing with that on the RHS of Eq. (1), in the sense of Ref.[13]. In the recent works of Jameson (Refs.[6, 7]) and Ta'asan, et al. (Ref.[8]), the adjoint equations they introduce are ultimately solved as discrete, incremental, iterative forms somewhat analogous to Eq. (9). However, the iterative details and construction of the adjoints differ from one another as well as what is outlined here.

The choice between using the sensitivity function (Q') or adjoint variable (A) approach in obtaining the single discipline sensitivity derivatives required for MdD via SA appears to depend upon whether there are a lesser number of active design variables (D) or a lesser number of required output functions (F), respectively. The sensitivity function Q' and the adjoint variable A are related, since, for $R^\star(Q, D, X) = 0$, $\tilde{F}(Q, D, X) = F(Q, D, X)$ according to Eq. (5) and upon subtracting Eq. (2) from Eq. (8) one obtains

$$\frac{\partial F}{\partial Q}Q' = A^T\left(\frac{\partial R^\star}{\partial D} + \frac{\partial R^\star}{\partial X}X'\right) \tag{10}$$

In a multi-point MdD exercise, both procedures may be used at different flow conditions, for example. In either case, though, the incremental or "delta-" form of the governing discrete equation set allows some leeway in the numerical algorithms which can be used to obtain efficient iterative solutions which are consistent discrete derivatives or adjoints. This form also appears extendable to 3-D CFD codes where computer memory limitations prevent solution by direct methods for the SD required in MdD.

CONSEQUENCES OF ITERATIVE NONCONVERGENCE

The previous sections have addressed the MdD requirements, implications, and consistency concerning SD from advanced CFD codes. The equations shown have been derived under the asumption that $R^\star(Q, D, X) = 0$, which is not generally true for iterative CFD solutions. In fact, for many "practical engineering results" obtained from 3–D CFD codes, one frequently accepts "solutions" $Q^{\star}$, where the maximum of R^n (over the entire flow field) may have been reduced only two or three orders of magnitude from its initial value, R^1. One associates the accuracy of $Q^{\star}$ with the discretization (truncation) errors due to the grid size and algorithm; this can be done only if the errors due to nonconvergence are not larger. The concept of consistent, discrete derivatives or adjoints, in the sense of Ref.[13], recognizes and admits the truncation error, but not others which may mask it.

Nonconvergence errors in $Q^{\star}$ show up in the coefficients of Eqs. (3) and (4) so that Q' inherits nonconvergence errors resulting from $R^n \neq 0$ as well as those from $\left(\frac{dR^{\star}}{dD}\right)^n \neq 0$. The Q' errors then propagate into the SD obtained from Eq. (2). For the adjoint variable formulation, it is seen from Eq. (5) that if $R^{\star} \neq 0$, then $\tilde{F} \neq F$. The solution for A from Eqs. (7) or (9) inherits errors in $Q^{\star}$ through the coefficient $\left(\frac{\partial R^{\star}}{\partial Q}\right)^T$; whereas the SD from Eq. (8), $\frac{d\tilde{F}}{dD}$, has errors due to those from both A and $Q^{\star}$ (through the partial derivatives of $R^{\star}$). Thus $\tilde{F} \neq F$ and $\frac{d\tilde{F}}{dD} \neq \frac{dF}{dD}$; there may be questions about which pair should be provided to the MdD: F and $\frac{d\tilde{F}}{dD}$ or $\tilde{F}$ and $\frac{d\tilde{F}}{dD}$.

OTHER IMPLICATIONS OF INCREMENTAL ITERATIVE FORM

The incremental iterative form for solution of large linear systems of equations, such as Eqs. (4) and/or (9), also leads to straightforward or better application of other computational methodologies. Of particular interest are aspects related to use of parallel processors and automatic differentiation (AD) techniques or other symbolic differentiation methods. In MdD via SA, several coarse-grained parallelizations occur naturally. For example, simultaneous analysis runs of codes from different disciplines and simultaneous analysis runs of a single discipline code at perturbed input conditions in order to obtain SD by "brute-force" finite differencing can be done in parallel. Here, however, we refer to fine-grained parallelization of codes to solve the incremental form equations for Q' and A. Again, the freedom allowed in the choice of approximate operators for the LHS of Eqs. (4) and (9) as opposed to those fixed operators on the LHS of Eqs. (3) and (7) can be exploited to tailor an algorithm to operate efficiently on a vector or massively parallel processor.

As pointed out by Griewank (Refs.[20, 21]), AD techniques have not yet been discovered by most applications-oriented code developers, i.e., the derivitive takers or users. Our naive perception is that AD is symbolic or exact algebraic differentiation at every uniary- or binary-operation code level combined with extensive use of the chain rule to compute the (forward) derivatives of a code output function with respect to its input variables. This is done before compilation, generating code such that upon execution these derivatives are calculated more-or-less in lockstep with the computation of that output function. Research on AD techniques continues (Ref.[22]) and some codes are available (Ref.[23]). Shubin (Ref.[14]) has suggested that these techniques might be useful for generating Jacobians, $\frac{\partial R^\star}{\partial Q}$, for CFD codes where they are not readily available in order to compute SD.

Our initial concerns about using AD to obtain Q' or A from Eqs. (3) or (7) were related to how the iterative implicit solution algorithm and its type-dependent operators would be "differentiated" and what the additional computational memory and time requirements would be. Preliminary unpublished results of Griewank and coworkers[2] show that a newly developed AD code, when applied to the iterative implicit solution algorithm (with its solution dependent operators) for the 2–D transonic small perturbation code of Elbanna and Carlson (Ref.[24]), produces consistent discrete derivatives. The point to be made here is that for Eqs. (4) or (9), AD needs to be applied only to get the RHS derivatives; the LHS operators are approximate and need not be exact Jacobians in order to get consistent discrete Q' or A. In either case, however, additional derivatives are also required for the RHS of Eqs. (2) or (8) in order to obtain the required SD. Issues regarding both computer run time and memory storage requirements in AD are still being actively investigated (See, for example, Ref. [22].). However, it appears that for equations cast in the iterative incremental form, such as Eqs. (4) or (9) as opposed to Eqs. (3) or (7), storage for only Jacobian-vector products and not the Jacobian matrix itself is required.

Elbanna and Carlson (Ref.[10]) have used symbolic differentiation methods to obtain the terms and coefficients required by an Eq. (3) form for the 3–D full potential equation on a Cartesian grid. These quantities are, of course, those appearing on the RHS of Eq. (4); perhaps this latter linear equation is the form to be solved, using an efficient approximate operator on the LHS.

Both the AD and symbolic differentiation techniques, when applied to the discretized code, should produce consistent discrete derivatives or adjoints. When either of these techniques reaches a robust routine applications stage then perhaps the quantities needed for the RHS of Eq. (4) (and/or Eq. (9)) can be generated from efficient Eq. (1) solvers; that is, Eq. (1) with approximate LHS operator $\widetilde{\frac{\partial R^n}{\partial Q}}$. Then SD can be calculated from those same efficient solvers with RHS terms appropriate to the derivatives and approximate LHS operators fixed at $Q^\star$.

CONCLUDING REMARKS

Several observations concerning various computational methodologies relevant to large-scale, multidisciplinary, gradient-based optimization for engineering systems design problems have been discussed. These observations address the situation where one (or more) discipline response(s) required by the optimized design procedure involve the analysis of a system of nonlinear PDE's which, when discretized, form a very large set of algebraic equations which must be solved iteratively. This situation occurs, for example, when advanced CFD codes are

[2] These findings were communicated by Andreas Griewank and coworkers, at Argonne National Laboratory. They applied their newly developed code ADIFOR, mentioned in the Preface of Ref.[22], to the referenced sample transonic flow analysis code provided to them by NASA.

applied in a multidisciplinary procedure for optimizing an aerospace vehicle design. A means for satisfying the MdD requirements for gradient information has been presented which appears to permit: first, some leeway in the CFD solution algorithms which can be used; second, an expansion to 3–D problems; and third, straightforward use of other computational methodologies. Many of these observations have been previously discussed as possibilities for doing parts of the problem more efficiently; the contribution here is observing how they fit together in a mutually beneficial way.

In particular, an incremental iterative form is proposed for solving the large systems of linear equations (for either the sensitivity functions Q' or adjoint variables A) needed in order to obtain consistent, discrete SD required by a MdD from 3–D CFD codes. This form, called the "delta" or "correction" form, is commonly used in CFD for obtaining the solution state vector Q from the nonlinear governing flow equations. It allows one to approximate the LHS operator of an iterative equation with one which can be adapted for efficiency, machine architecture, computational algorithm, etc.; all that is required of the approximation is convergence. What appears on the RHS of the iterative equation is the current iterate evaluation of the condition to be satisfied; that is, a zero at convergence. It is also noted that symbolic differentiation or AD techniques may be used to create, from existing CFD codes, the derivative terms needed to construct the RHS of this incremental iterative equation form from which computation of consistent, discrete SD follows.

APPENDIX — INDICIAL DEPENDENCIES OF VARIABLES WITH DIMENSIONAL SIZE AND COMPUTATIONAL ESTIMATES

One purpose of this appendix is to indicate the dependency of the variables introduced in the main text upon relevant indicial parameters in order to estimate their vector or matrix sizes. These parameters are governed not only by the specific physical problem and its modelling, but also by both the CFD solution technique and the optimized MdD procedure. It is assumed here that the fluid flow problem to be solved by CFD is three dimensional, since this will be the case for most MdD exercises for aerospace vehicles.

Another purpose of this appendix is to estimate the magnitude of the CFD task in terms of computational time and number of required solutions. Several indices are included in order to estimate the total number of solutions required in the MdD process.

Discipline Variables

The variables introduced in the text and listed below occur (that is, can be defined or may be used) in each of the individual disciplines which participate in a MdD exercise. These variables are illustrated here by indicating typically what they are in steady aerodynamics.

A — discrete adjoint variable; constructed with respect to a required output function, F. See Eq. (7) in text.

D — input design variable. Typically, for aerodynamics, the design variables are parameters which describe the allowed changes in vehicle surface shape and, perhaps, flow conditions during the design optimization cycles; that is, parameters affecting inner and, perhaps, outer boundary conditions for both the CFD and grid generation Boundary Value Problems (BVP's).

F — required output function; needed by either another discipline analysis code or the MdD optimization procedure. Typically, for aerodynamics, the output functions are quantities such as surface loadings, force and moment coefficients, stability derivatives, etc. at several (perhaps even a number of) flight conditions.

Q — solution state vector; required unknown flow variables associated with each discrete volume grid point.

$Q' = \frac{\partial Q}{\partial D}$ — sensitivity function of the solution state vector, Q, with respect to a design variable, D.

R — discrete representation of discipline governing equation; for aerodynamics, this is the residual vector associated with each flow equation at the discrete volume grid points.

X discrete volume grid point vector; three components for each grid point.

$X' = \frac{\partial X}{\partial D}$ sensitivity of the discrete volume grid point vector, X, with respect to a design variable, D. For surface-fitted grids typical of advanced CFD codes, $X' \neq 0$; in fact, it is not even necessarily sparse.

Description of Indices

Indices or labels are associated with each of the discipline variables given above; and said indices are described here.

a discipline index; typically max(a) is several (disciplines such as aerodynamics, structures, performance, controls, weights, etc.).

d design variable index; max(d) depends on problem and discipline.

f output function index; max(f) depends on problem and discipline. For some disciplines output functions are required at several (or a number of) flight conditions such as for load cases and polars (See index s below.).

i , j , k three-dimensional (volume) grid point indices; max(i), max(j), and max(k) depend on configuration complexity, flow equation approximation, and spatial resolution required.

i", j" two-dimensional (surface) grid point indices; max(i") and max(j") each typically range from 50% to 65% of max(i) and max(j), respectively.

m function evaluation index for solutions required by the optimizer at each outer loop; max(m) varies from one to ten times several, depending upon the optimizer.

n iteration or time-step index for CFD solution; max(n) depends on algorithm and convergence level required.

o iteration index for outer optimizer loop of MdD; typically max(o) is several to ten times several.

q unknown variable index for components of solution state vector Q at each volume grid point; max(q) depends on flow equation approximation.

s function evaluation index for solutions at different flow or load conditions for each outer optimizer loop; max(s) from several to 10 times several.

t single iteration execution time per grid point; depends on algorithm, computer, and grid size(with its dependencies).

τ average time required to generate solution for "single" SD; depends on the method used.

Size Estimates for Indices

Size estimates for typical max values required for indices defined above depend, of course, on many factors; thus, some assumptions must be made. The number of design variables, max(d), and the number of required output functions, max(f), depend upon the discipline and MdD problem. The estimates here are made for the steady external aerodynamics CFD discipline and a wing-fuselage-vertical tail semi-span HSCT configuration with engines. Note that this engine simulation is for external airframe/engine integration effects on design; it is not for the internal propulsion CFD simulation effects nor engine design. In the MdD environment it appears to us that max(f) will be about the same size as max(d), since detail pressure loads will have to be supplied to the structures discipline and polars to the performance discipline. Thus, max(f)$\sim$max(d)$\sim$100 to several times 100.

For CFD calculations at sub-, trans-, and supersonic flow conditions required for the HSCT design, iterative Euler or Navier-Stokes codes are required. Thus, max(q) = 5. Estimates for max(i x j x k), max(t), and max(n) as well as central memory (CM) and solution time (T) requirements are given in Table 1. These estimates have been extrapolated from solutions run at $\frac{1}{2}$ to $\frac{3}{4}$ million grid points in codes which are among the most efficient presently available; these 3–D codes appear to achieve multigrid (MG) convergence rates (i.e., proportional to the total number of grid points).

Table 1— Estimates of Computer Run Time and Memory Storage Requirements for 3–D Configuration Analysis from Several Advanced CFD Codes Which Use Multigrid Acceleration Techniques.

Code	Eqn.	Ref.	max (i x j x k)	max(t) (sec*/gp/iter)	max (n)†	T (hr*)	CM (Mw)
TLNS3D	Euler	[25], ‡	1.5 x 10^6	40 x 10^{-6}	250	4.2	74
(Single block)	TLRANS		3.0 x 10^6	60 x 10^{-6}	350	18	160
EUL3D	Euler	[26, 27], •	1.5 x 10^6	150 x 10^{-6}	250	16	180
(Unstructured)							
TNS3DMB	Euler	§	1.5 x 10^6	130 x 10^{-6}	250	14	77
(Multiblock)	TLRANS		3.0 x 10^6	200 x 10^{-6}	350	58	140

∗ Single processor Cray 2 time (LaRC Voyager).

† n for reduction of $\frac{R^n}{R^1}$ to about 10^{-4}.

‡ Estimates provided by code developer: V.N. Vatsa of NASA, LaRC.

• Estimates provided by code developer: D.J. Mavriplis of ICASE, NASA, LaRC.

§ Estimates provided by code developers: V.N. Vatsa and M.D. Sanetrik of NASA, LaRC.

The estimates given in Table 1 are not in disagreement with those quoted in Ref.[3] (See their Tables 4. and 5.). The codes listed here are, however, among the fastest listed in Ref.[3]. This, coupled with a perhaps optimistic estimate of the required max(i x j x k) gives minimal estimates for some of the quantities of interest.

Dependence of Variables on Indices

Strictly speaking, each of the discipline variables previously listed carries the individual discipline index, a, as well as the outer optimizer loop index, o. This latter index, however, may appear explicitly only on the functions D, F, and $\frac{dF}{dD}$; that is, those discipline input (design) and output functions and corresponding sensitivity derivatives seen by the optimizer. Dependencies of the variables on the other indices is given in Table 2, where the notation v means vector, m means matrix, whereas c and r are column and row indices, respectively.

Table 2— Indicial Dependence of Variables

Variable	Type	Index				
		d	f	i x j x k	3^*	q
A	m		c	r		r
D	v	r				
F	v		r			
Q	v			r		r
Q'	m	c		r		r
R	v			r		r
X	v			r	r	
X'	m	c		r	r	

∗ indicates 3-D space coordinates of a grid poiint.

Dimensional Size Estimates for Variables

The length of a vector is the product of the max(indices) denoted r in Table 2; that is, for example, the length of the vector Q is max(i x j x k) x max(q). The size of a matrix is then the product of max(indices) denoted r x max(index) denoted c. As already noted, estimates for the maximum sizes of indices depending on the flow solution are governed by configuration complexity, flow equation approximation, CFD solution technique, and required spatial resolution and iterative convergence. Other indices depend upon the design problem and methods used in the optimization. The computational molecule (stencil) for typical 3–D CFD codes is order 10 so that Jacobian matrices, $\frac{\partial R}{\partial Q}$, are sparse (only a few % dense) and have a banded, nonsymmetric structure. Table 3 summarizes estimated dimensional sizes of discipline variables, based upon estimates in previous sections.

Table 3— Estimates for Dimensional Size of Variables

Variables	**Type**	**Size**
D, F	v	10^2
Q, R, X	v	10^7
$A, Q', X', \frac{dR}{dD}$	m	10^7 x 10^2
$\frac{\partial R}{\partial Q}, \frac{\partial R}{\partial X}$	m	10^7 x 10^7
$\frac{dF}{dD}$	m	10^2 x 10^2
$\frac{\partial F}{\partial Q}, \frac{\partial F}{\partial X}$	m	10^2 x 10^7

Computational Task Estimates

The total CFD computational task for steady external aerodynamics in a MdD is dependent upon a number of quantities already estimated. Many of the output functions F (such as the detail pressure loadings required by the structures discipline) are generated in a single CFD code run; however, several (to many) load conditions in the flight envelope are required. This is also true for the performance discipline's need for polars. The multiplicative factor for such solutions is max(s). The number of solution evaluations needed by an optimizer in a single cycle is max(m), whereas the number of optimizer cycles is max(o). Thus the estimated number of function evaluations, N_F, (not including those for SD calculations) required is N_F = max(o) x max(m) x max(s) or using the upper and lower max estimates

$$10 \leq N_F \leq 10^{4.5} \tag{11}$$

For the SD calculation one must first solve for either max(d) sensitivity functions Q' or max(f) adjoint variables A at each optimizer step and load or flight condition. Then the estimated number of SD evaluations, $N_{F'}$, required is $N_{F'}$ = max(o) x max(s) x $\left\{ \begin{array}{c} \text{max(d)} \\ \text{or} \\ \text{max(f)} \end{array} \right\}$ or using the upper and lower max estimates

$$10^3 \leq N_{F'} \leq 10^{5.5} \tag{12}$$

The estimated total computational time for steady external CFD in a complete MdD, T_T, is then

$$\begin{aligned} T_T &= N_F \text{ x } T + N_{F'} \text{ x } \tau \\ &= \left(1 + (N_{F'}/N_F)\left(\frac{\tau}{T}\right)\right) N_F \text{ x } T \\ &\sim \left(1 + 100\left(\frac{\tau}{T}\right)\right) N_F \text{ x } T \end{aligned} \qquad (13)$$

where the last step follows approximately from Eqs. (11) and (12). In order to get T_T to a reasonable number, it is clear that $\left(\frac{\tau}{T}\right)$ and N_FxT must be made as small as possible. The present incremental iterative form addresses getting $\left(\frac{\tau}{T}\right)$ smaller than one, the approximate value for "brute-force" finite difference determination of the SD. Other approximation techniques, such as discussed in Refs.[11, 12], or other design optimization strategies, as in Refs.[6–8], for example, must be used to decrease N_F x T.

REFERENCES

1. S. Dollyhigh and J.S-. Sobieski. Recent Experience with Multidisciplinary Analysis and Optimization in Advanced Aircraft Design. In *Third Air Force/NASA Symposium On Recent Advances in Multidisciplinary Analysis and Optimization, A Collection of Technical Papers*, pages 404–411, San Francisco, CA, September 1990.
2. P. G. Coen, J. S-. Sobieski, and S. Dollyhigh. Preliminary Results from the High-Speed Airframe Integrated Research Project. AIAA Paper No. 92–1004, February 1992.
3. T. L. Holst, M. D. Salas, and R. W. Claus. The NASA Computational Aerosciences Program—Toward Teraflops Computing. AIAA Paper No. 92–0558, January 1992.
4. A. Jameson. Successes and Challenges in Computational Aerodynamics. AIAA Paper No. 87–1184–CP, June 1987.
5. G. S. Dulikravich. Aerodynamic Shape Design and Optimization. AIAA Paper No. 91–0476, January 1991.
6. A. Jameson. Aerodynamic Design via Control Theory. *Journal of Scientific Computing*, 3:233–260, 1988. (Also NASA CR–181749 or ICASE Report No. 88-64, November 1988).
7. A. Jameson. Automatic Design of Transonic Airfoils to Reduce the Shock Induced Pressure Drag. Princeton Univ. MAE Report 1881, 1990. (Also 31st Israel Annual Conference in Aviation and Aeronautics, February 1990).
8. S. Ta'asan, G. Kuruvila, and M. D. Salas. Aerodynamic Design and Optimization in One Shot. AIAA Paper No. 92-0025, January 1992.
9. J. S-. Sobieski. Multidisciplinary Optimization for Engineering Systems: Achievements and Potential. NASA TM 101566, March 1989.
10. H. M. Elbanna and L. A. Carlson. Determination of Aerodynamic Sensitivity Coefficients Based on the Three-Dimensional Full Potential Equation. AIAA Paper No. 92–2670, June 1992.
11. A. C. Taylor III, G. J-. W. Hou, and V. M. Korivi. Sensitivity Analysis, Approximate Analysis, and Design Optimization for Internal and External Flows. AIAA Paper No. 91–3083, September 1991.
12. O. Baysal, M. E. Eleshaky, and G. W. Burgreen. Aerodynamic Shape Optimization Using Sensitivity Analysis on Third-Order Euler Equations. AIAA Paper No. 91–1577–CP, June 1991.
13. G. R. Shubin and P. D. Frank. A Comparison of Two Closely-Related Approaches to Aerodynamic Design Optimization. In *Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences(ICIDES-III)*, pages 67–78, Washington, D.C., October 1991. (Also Technical Report AMS-TR-163, Boeing Computer Services, April 1991).
14. G. R. Shubin. Obtaining "Cheap" Optimization Gradients from Computational Aerodynamics Codes. Technical Report AMS-TR-164, Boeing Computer Services, June 1991.
15. A. C. Taylor III, V. M. Korivi, and G. J-. W. Hou. Sensitivity Analysis Applied to the Euler Equations: A Feasibility Study with Emphasis on Variation of Geometric Shape. AIAA Paper No. 91–0173, January 1991.
16. R. E. Smith and I. Sadrehaghighi. Grid Sensitivity in Airplane Design. In *4th International Symposium on Computational Fluid Dynamics*, volume II, pages 1071–1076, Univ. of Calif., Davis, September 1991.

17. J. S-. Sobieski. The Case for Aerodynamic Sensitivity Analysis. NASA CP-2457, pages 77–96, January 1987.

18. V. M. Korivi, A. C. Taylor III, P. A. Newman, G. J-. W. Hou, and H. E. Jones. An Incremental Strategy for Calculating Consistent Discrete CFD Sensitivity Derivatives. NASA TM 104207, February 1992.

19. G. J-. W. Hou, A. C. Taylor III, and V. M Korivi. Discrete Shape Sensitivity Equations for Aerodynamic Problems. AIAA Paper No. 91–2259, June 1991.

20. A. Griewank. The Chain Rule Revisited in Scientific Computing. *SIAM News*, May 1991.

21. A. Griewank. The Chain Rule Revisited in Scientific Computing, Part II. *SIAM News*, July 1991.

22. A. Griewank and G. F. Corliss, editors. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*. SIAM, 1991.

23. D. W. Juedes. A Taxonomy of Automatic Differentiation Tools. In *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, pages 315–330. SIAM, 1991.

24. H. M. Elbanna and L. A. Carlson. Determination of Aerodynamic Sensitivity Coefficients in the Transonic and Supersonic Regime. AIAA Paper No. 89–0532, January 1989.

25. V. N. Vatsa and B. W. Wedan. Development of a Multigrid Code for 3-D Navier-Stokes Equations and Its Application to a Grid-Refinement Study. *Computers & Fluids*, 18(4):391–403, 1990.

26. D. J. Mavriplis. Three Dimensional Unstructured Multigrid for the Euler Equations. NASA CR 187565, May 1991.

27. R. Das, D. J. Mavriplis, J. Saltz, S. Gupta, and R. Ponnusamy. The Design and Implementation of a Parallel Unstructured Euler Solver Using Software Primitives. AIAA Paper No. 92–0562, January 1992.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY ***(Leave blank)***

2. REPORT DATE
February 1992

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Observations on Computational Methodologies for Use in Large-Scale
Gradient-Based, Multidisciplinary Design Incorporating Advanced CFD Codes

5. FUNDING NUMBERS
505-59-53-01

6. AUTHOR(S)
P. A. Newman; G. J.-W. Hou; H. E. Jones; A. C. Taylor III; and V. M. Korivi

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center, Hampton, VA 23665-5225;
Aerostructures Directorate, U.S. Army-AVSCOM, Langley Research Center,
Hampton, VA 23665-5225

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration, Washington, DC 20546

U.S. Army Aviation Systems Command, St. Louis, MO 63120-1798

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA TM-104206
AVSCOM TR- 92-B-007

11. SUPPLEMENTARY NOTES
Newman: Langley Research Center, Hampton, VA; Hou: ODU, Norfolk, VA; Jones: Aerostructures Directorate, USA-AVSCOM , Hampton, VA; Taylor: ODU, Norfolk, VA; Korivi: ODU, Norfolk, VA

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified - Unlimited

Subject Category 02

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

This paper briefly outlines how a combination of various computational methodologies could reduce the enormous computational costs envisioned in using advanced CFD codes in gradient-based optimized multidisciplinary design (MdD) procedures. Implications of these MdD requirements upon advanced CFD codes are somewhat different than those imposed by a "single" discipline design. A means for satisfying these MdD requirements for gradient information is presented which appears to permit: first, some leeway in the CFD solution algorithms which can be used; second, an extension to 3-D problems; and third, straightforward use of other computational methodologies. Many of these observations have previously been discussed as possibilities for doing parts of the problem more efficiently; the contribution here is observing how they fit together in a mutually beneficial way.

14. SUBJECT TERMS
Sensitivity Analysis
Consistent Sensitivity Derivatives

15. NUMBER OF PAGES
18

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT